

第一章 Multi Infineon 介绍

Multi Infineon 产品介绍.....3

第二章 开发系统安装

硬件安装.....5

第三章 软件安装

WINDOWS 版本软件安装.....8

编译器安装.....8

第四章 开发环境

菜单 | 文件.....9

菜单 | 编辑.....13

菜单 | 搜索.....13

菜单 | 项目.....14

菜单 | 执行.....14

菜单 | 窗口.....15

菜单 | 外设.....22

菜单 | 仿真器.....22

快速入门26

伟福文本编辑器使用31

XC166 用户板.....34

附录：XC166 用户板电路图.....41

XC866 用户板.....44

附录：XC866 用户板电路图.....47

XC866 串口下载.....48

1 概述

1

尊敬的用户：

您好！非常感谢您使用伟福系列仿真器。伟福仿真品种多、功能强，和国内外同类高档仿真器功能相比，软、硬件方面具有多种先进特点。

硬件方面先进的特点：

1. 高速 JTAG 口仿真器。
2. 可以仿真 XC166 系列 16 位单片机及 XC866 系列系列 8 位单片机
3. 自动适应用户板电压，范围 2.5V-5.5V
4. 强 JTAG 口驱动，允许 JTAG 口串接多个芯片

软件方面先进的特点：

WAVE866 IDE 环境，中/英文界面可任选，用户源程序的大小不再有任何限制。有丰富的窗口显示方式，多方位，动态地展示仿真的各种过程，使用极为便利。

双工作模式： a) 软件模拟仿真（不用仿真器也能模拟运行用户程序）。b) 硬件仿真。

真正集成调试环境： 集成了编辑器、编译器、调试器，源程序编辑、编译、下载、调试全部可以在一个环境下完成。

项目管理功能： 现在单片机软件越来越大，也越来越复杂，维护成本也很高，通过项目管理可化大为小，化繁为简，便于管理。项目管理功能 也使得多模块，多语言混合编程成为可能。

多语言多模块混合调试： 支持汇编、C 语言、PL/M 语言多模块混合源程序调试，在线直接修改、编译、调试源程序。如果源程序有错，可直接定位错误所在行。

直接点屏观察变量： 在源程序窗口，点击变量就可以观察此变量的值，方便快捷。

功能强大的变量观察： 支持 C 语言的复杂类型，树状结构显示变量，

强大的书签、断点管理功能：书签、断点功能可快速定位程序，为编写、查找、比较程序提供帮助。

类似 IE 的前进、后退定位功能：可以在项目内跨模块地定位光标前一次或后一次位置，为比较、分析程序提供帮助。

类似 Delphi 的界面操作：类似 Delphi 的集成调试环境，灵活多变的窗口“靠岸” (Docking) 功能，可以方便地将窗口平排靠岸，或以页面方式靠岸，任由用户自己安排。桌面整洁，操作灵活。

方便实用、功能多样的源程序编辑窗口：a.) 窗口分隔功能可将源程序窗口分成两个完全独立的编辑窗口，而所编辑的内容却是同一程序，为分析、比较检查大程序提供方便。b.) 语法相关彩色显示，使得编写程序轻松，观察程序醒目。且用户可自己定义所喜好颜色，享受个性化编程带来的快乐 c.) 书签功能提供多达 9 个书签，使得您在分析、比较、检查大程序时从容不迫。d.) 寻找配对符号功能为您在复杂程序嵌套中找到“另一半”，例如可以找与‘{’相对的‘}’，或为‘(’找到相对的‘)’。e.) 多行程序的同进同退功能，可以使得程序错落有致，帮您编写优美、整洁的程序。

外设管理功能：外设管理可以让您在调试程序时，观察到端口、定时器、串行口中断、外部中断相关的寄存器的状态，更可以帮您完成这些外设的初始化程序，包括 C 语言和汇编语言，而您所做的只是填表，定义外设所要完成的功能。

南京伟福实业有限公司

电话：(025) 84727244, 84727344

传真：(025) 84727349

电邮：wave-cn@263.net

网址：<http://www.wave-cn.com>

2 硬件安装

Multi Infineon 物品清单：

- ① Multi Infineon 硬件仿真器
- ② 16 针标准 Multi Infineon 接口扁平连接线缆一条
- ③ 并口连接线缆一条
- ④ 直流 5V 开关电源一个
- ⑤ WAVE 使用手册光盘

外形示意图



2.1 25 芯并口

一端连接 Multi-Infineon 仿真器，另一端连接 PC 机的并口

注意：要将打印机的并口工作方式设置为：EPP 或 ECP

2.2 电源

Multi-Infineon 使用单+5V 电源，仿真器所配电源仅供仿真器本机使用，

目标板应使用另外的电源，目标板电压范围为 2.5V 至 5.5V。

2.3 16 芯仿真电缆

与目标板连接的 JTAG 仿真电缆，红线为 1 脚。信号定义见第 25 页

2.4 指示灯

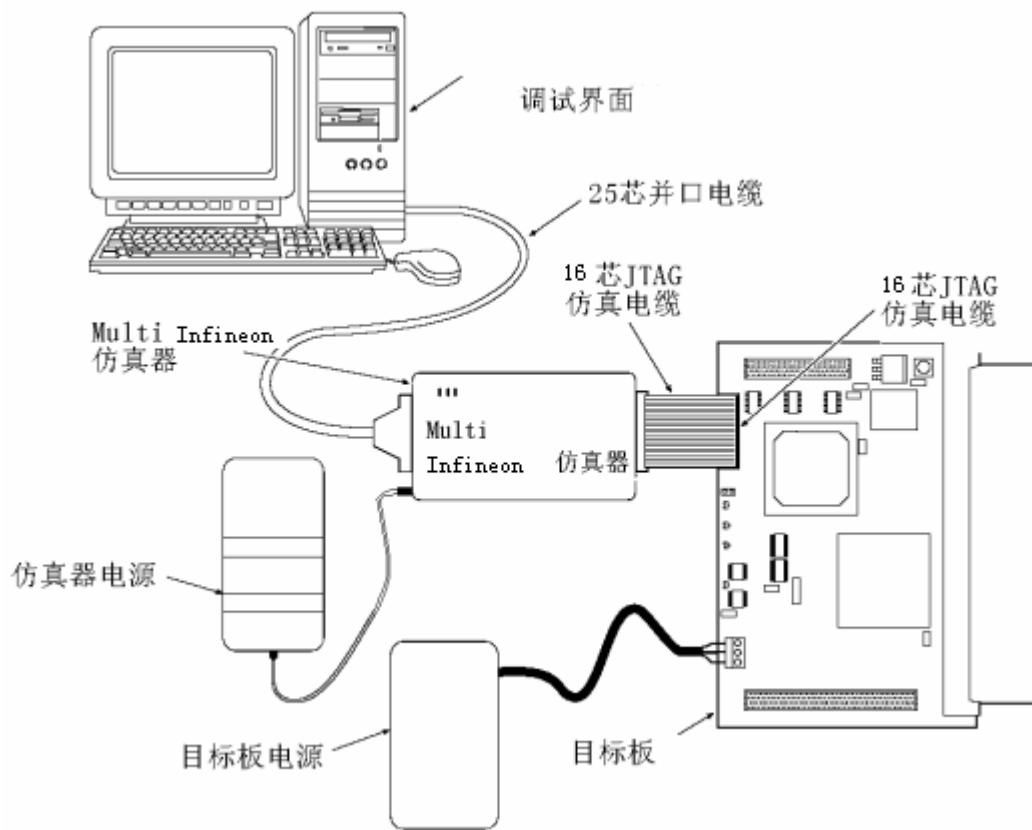
Multi-Infineon 有三个指示灯

红色：电源指示。当正确接入电源后该指示灯亮。

当电源故障（过压、过流、极性接反）时，仿真器内的保护电路会自动断开电源。该指示灯灭。

绿色：运行指示。当 Multi-Infineon 仿真器与 PC 机通信时，该指示灯亮闪烁

黄色：状态指示。在加电时，仿真器首先进行自检，此时该指示灯快速闪烁，当仿真器无法通过自检时，该指示灯常亮，表明仿真器有故障。此后每隔 20 秒仿真器会再次进行自检。当仿真器通过自检后，仿真器会进一步检查目标板是否连接，若目标板连接正确，该指示灯灭，否则该指示灯每秒闪烁一次。



按上图所示连接好PC、Multi Infineon仿真器、目标板，并将仿真器及目标板电源打开。此时应红色电源指示灯亮，黄色状态指示灯灭。表明仿真器及目标板已连接好。

3 软件安装

WINDOWS 版本软件安装

- 1. 将光盘放入光驱，光盘会自动运行，出现安装提示。
- 2. 按照安装程序的提示, 输入相应内容。
- 3. 继续安装, 直至结束.

在安装过程 XC166 中, 如果用户没有指定安装目录, 安装完成后, 会在 C: 盘建立一个 C:\WAVE166 目录(文件夹), 结构如下:

目录	内容
C:\WAVE166	
└ BIN	可执行程序及相关配置文件
└ HELP	帮助文件和使用说明
└ SAMPLES	样例和演示程序

在安装过程 XC866 中, 如果用户没有指定安装目录, 安装完成后, 会在 C: 盘建立一个 C:\WAVE866 目录(文件夹), 结构如下:

目录	内容
C:\WAVE866	
└ BIN	可执行程序及相关配置文件
└ HELP	帮助文件和使用说明
└ SAMPLES	样例和演示程序

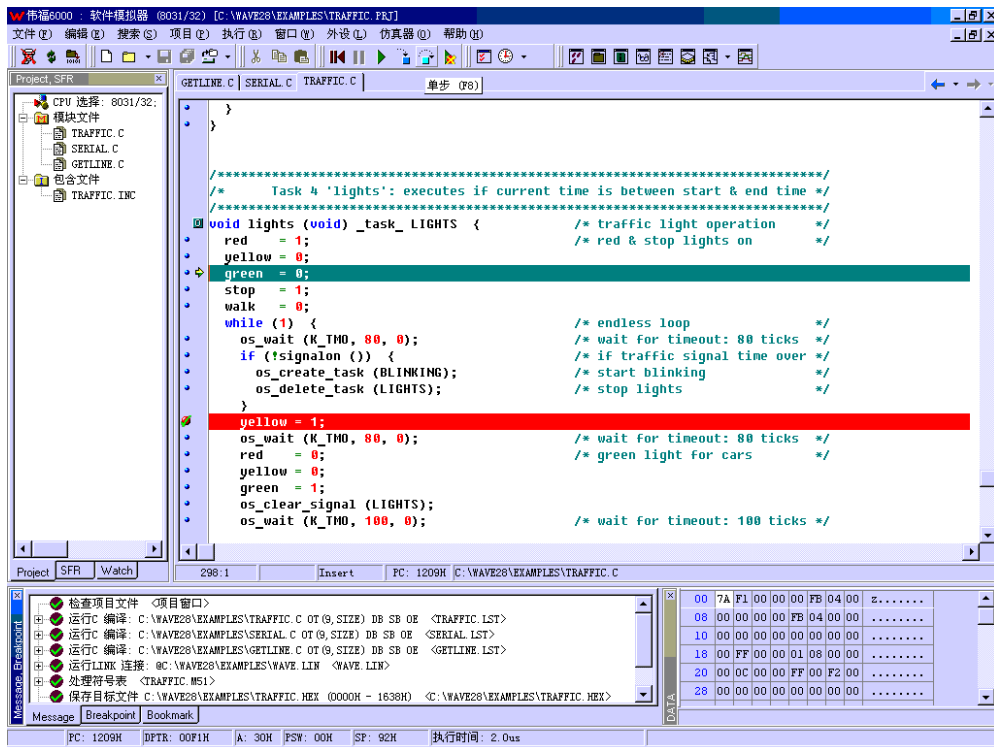
可以从公司网站下载软件的最新版本。网站网址为 <http://www.wave-cn.com>

编译器安装

伟福仿真系统已留有第三方的编译器的接口, 方便用户使用高级语言调试程序. 编译器请用户自备.

- 例如: “C:\KEIL\C166\”
- 例如: “C:\KEIL\C51\”

4 开发环境



文件(F)

文件 | 打开文件

打开用户程序，进行编辑。如果文件已经在项目中，可以在项目窗口中双击相应文件名 打开文件。

文件 | 保存文件

保存用户程序。用户在修改程序后，如果进行编译，则在编译前，系统会自动将修改过的文件存盘。

文件 | 新建文件

建立一个新的用户程序，在存盘的时候，系统会要求用户输入文件名。

文件 | 另存为

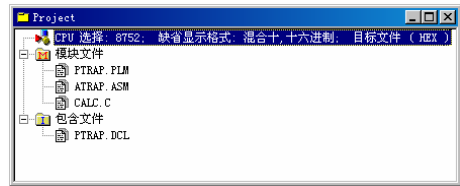
将用户程序存成另外一个文件，原来的文件内容不会改变

文件 | 重新打开

在重新打开的下拉菜单中有最近打开过的文件及项目，选择相应的文件名或项目名就可以重新打开文件或项目。

文件 | 打开项目

打开一个用户项目，在项目中，用户可以设置仿真类型。加入用户程序，进行编译，调试。系统中只允许打开一个项目，打开一个项目或新建一个项目时，前一项目将自动关闭。



(图：项目窗口)

伟福开发环境的项目文件包括仿真器设置, 模块文件, 包含文件。

仿真器设置包括仿真器类型, 仿真头 (POD) 类型, CPU 类型, 显示格式和产生的目标文件类型可以用以下几种方法设置仿真器。

- o 在项目窗口中双击第一行, 将打开仿真器设置窗口, 对仿真器进行设置。
- o 按鼠标右键, 在弹出菜单中选择[仿真器设置]。
- o 主菜单 仿真器|仿真器设置。

加入模块文件 o 按鼠标右键, 在弹出菜单中选择[加入模块文件]

- o 主菜单 项目|加入模块文件

加入包含文件

- o 按鼠标右键, 在弹出菜单中选择[加入包含文件]
- o 主菜单 项目|加入包含文件

用户可以将以前单文件方式仿真转为 WINDOWS 下的项目方式进行仿真

1. 主菜单 文件|新建项目, 在新建项目时, 前一个项目自动关闭。
2. 加入模块文件时, 选择要调试的程序文件名, 将文件加入项目。
3. 将项目存盘。
4. 编译, 运行, 调试项目。

文件 | 保存项目

将用户项目存盘。用户在编译项目时, 自动存盘。 注意:

当用项目仿真时, 系统要求项目文件, 模块文件, 包含文件在同一个目录(文件夹)下。

文件 | 新建项目

当用户开始新的任务时, 应新建一个项目, 在项目中, 设置所用仿真器类型, POD 类型, 加入用户程序 (模块)。

文件 | 关闭项目

关闭当前项目, 如果用户不想用项目方式调试单个程序, 就要先关闭当前项目。

文件 | 项目另存为

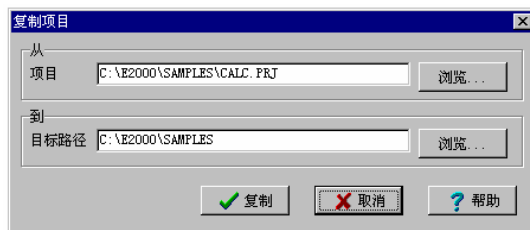
将项目换名存盘,此方法只是将项目用另一个名字,而不会将项目中的模块和包含文件换成另一个名字存盘.如果想将整个项目及模块存到另一个地方,请用复制项目方法.

文件 | 复制项目

复制项目,用户可以将项目中的所有模块(用户程序)备份到另一个地方.在多模块项目中,用复制项目功能,可以避免用户因为少复制某些模块,而造成项目编译不能通过.方便用户对程序进行管理.

复制项目对话框.

[从项目]栏中为当前被复制项目,包括项目中各模块,包含文件,如果不是复制当前项目,可以通过[浏览]找到所要复制的项目,[到目标路径]中为项目复制到何处,可以通过其后的[浏览]指定将项目复制到其它地方.



文件 | 调入目标文件

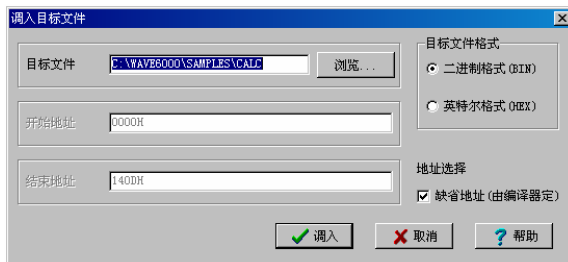
装入用户已编译好目标文件。系统支持两种目标文件格式：BIN，HEX 格式

调入已经编译好的目标文件

目标文件格式有二种：

二进制 (BIN) : 由编译器生成的二进制文件,也就是程序的机器码

英特尔格式 (HEX) : 由英特尔定义的一种格式,用 ASCII 码来存储编译器生成的二进制代码,这种格式包括地址,数据和校验



(图：调入目标文件对话框)

地址选择一般为缺省地址(由编译器定). 如果想在当前项目已编译好的二进制代码中插入一段其它代码,可以去掉"缺省地址"前的选择. 然后填入开始插入的地址和结束地址. 用调入目标文件的方法,可以调试已有的二进制代码程序. 而不需要源程序.

直接调入目标文件进行仿真的方法是：

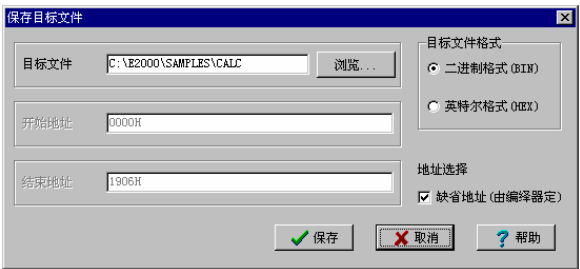
1. 关闭项目。
2. 在新建的项目中, 设置仿真器类型, 仿真头类型, CPU 类型.
3. 调入目标文件. (不要用加入模块方式, 而是直接调入文件)

- 4. 打开 CPU 窗口, 在 CPU 窗口中就可以看见目标文件反汇编生成的程序.
- 5. 程序停在与 CPU 相关的地址上(XC866 系列停在 0000H 处, XC166 系列停在 C00000H)
- 6. 这样就可以单步或全速调试程序了.

文件 | 保存目标文件

将用户编译生成的目标文件存盘。

对于按项目方式仿真的用户, 系统将程序编译正确后, 会根据用户在 仿真器 | 仿真器设置下[目标文件] 中设置的格式, 将生成的二进制代码存盘. 如果用户是用调入目标文件方式进行仿真, 并且对目标码进行了修改, 就可以用 文件|保存目标文件 方式存盘.



目标文件可以存成两种格式：

- [**二进制格式 (BIN)**]: 由编译器生成的二进制文件, 也就是程序的机器码
- [**英特尔格式 (HEX)**]: 由英特尔定义的一种格式, 用 ASCII 码来存储编译器生成的二进制代码, 这种格式包括地址, 数据和校验
- [**地址选择**]一般为[缺省地址(由编译器定)]. 如果想要存盘的目标文件是由“调入目标文件”方式装入, 而不是由系统编译产生的代码, 并已经修改, 最好指定它的开始地址和结束地址, 因为代码不是编译系统产生的. 系统不知道文件有多长, 无法指定开始和结束地址。自己指定地址的方法是：去掉[缺省地址]前的选择勾, 然后填入开始插入的地址和结束地址。

文件 | 反汇编

将可执行的代码反汇编成汇编语言程序。（详见伟福反汇编功能的使用方法）

文件 | 打印

打印用户程序。

文件 | 退出

退出系统, 如果在退出以前有修改过的文件没有存盘, 系统将会提示是否把文件存盘.

编辑(E)

- 编辑 | 撤消键入
取消上一次操作
- 编辑 | 重复键入
恢复被取消的操作
- 编辑 | 剪切
删除选定的正文，删除的内容被送到剪贴板上
- 编辑 | 复制
将选定的内容，复制到剪贴板上
- 编辑 | 粘帖
将剪贴板的内容插入光标位置
- 编辑 | 全选
选定当前窗口所有内容。

搜索(S)

- 搜索 | 查找
在当前窗口中查找符号，字串。可以指定区分大小写方式，全字匹配方式，可以向上 / 向下查找。
- 搜索 | 在文件中查找
可以在指定的一批文件中查找某个关键字。
- 搜索 | 替换
在当前窗口查找相应文字，并替换成指定的文字，可以指定区分大小写方式和全字匹配方式查找，可以在指定处替换，也可以全部替换。
- 搜索 | 查找下一个
查找文字符号下一次出现的地方
- 搜索 | 项目中查找
在项目所有模块（文件）中查找符号，字串。在项目所包含的文比较多时，用此方法可以很方便地查到字串在什么地方出现。
- 搜索 | 转到指定行
将光标转到程序的某一行。
- 搜索 | 转到指定地址/标号
将光标转到指定地址或标号所在的位置。
- 搜索 | 转到当前 PC 所在行
将光标转到 PC 所在的程序位置。

项目 (P)

项目 | 编译

编译当前窗口的程序。如有错误，系统将会指出错误所在的位置。

项目 | 全部编译

全部编译项目中所有的模块（程序文件），包含文件。如有错误系统会指出错误所在位置。

项目 | 装入 OMF 文件

建好项目后，无须编译，直接装入在其它环境中编译好的调试信息，在伟福环境中调试。

项目 | 加入模块文件

在当前项目中添加一个模块。

项目 | 加入包含文件

在当前项目中添加一个包含文件

执行 (R)

执行 | 全速执行

运行程序

执行 | 跟踪

跟踪程序执行的每步，观察程序运行状态。

执行 | 单步

单步执行程序，与跟踪不同的是，跟踪可以跟踪到函数或过程的内部，而单步执行则不跟踪到程序内部。

执行 | 执行到光标处

程序从当前 PC 位置，全速执行到光标所在的行。如果光标所在行没有可执行代码。则提示“这行没有代码”

执行 | 暂停

暂停正在全速执行的程序。

执行 | 复位

终止调试过程，程序将被复位。如果程序正在全速执行，则应先停止。

执行 | 设置 PC

将程序指针 PC，设置到光标所在行。程序将从光标所在行开始执行。

执行 | 自动单步跟踪/单步

模仿用户连续按 F7 或 F8 单步执行程序。

执行 | 编辑观察项

观察变量或表达式的值，可以将需要检查和修改的值或表达式放到观察窗口里以便检查和修改。（图：观察项对话框）



[表达式]：用于输入用户所要求值的表达式。

[重复次数]：如果表达式为某一存储变量，重复次数表示以此变量开始的连续 N 个地址的值。

[显示格式]：指定用何种方式显示表达式的值。

[存储区域]：指明变量所在的区域。

[显示类型]：指定表达式为何种类型的变量。

[缺省方式显示]：按照高级语言定义的方式显示。

[存储器内容]：以内存方式显示观察内容，也就是按地址顺序显示变量值，与变量类型无关

[求值]：对表达式求值，并按显示格式显示在窗口内。

[加入观察]：将表达式加入观察窗口中，以便随时察看。

[编辑观察]：当修改过窗口内容后，按此键后，替代观察窗口中的原观察项，如果选择[加入观察]，则会在观察窗口中另加一个变量的观察项，以两种格式观察一个变量。

[取消]：关闭编辑观察项窗口

执行 | 设置/取消断点

将光标所在行设为断点，如果该行原来已为断点，则取消该断点。所有断点通过断点窗口进行管理。

四种方法可以在光标处设置断点

1. 将光标移到编辑窗口内, 行左边的空白处, 光标变成“手指圆”箭头, 单击鼠标左键, 可以设置/取消断点.
2. 使用 Ctrl-F8 快捷键, 可以在光标所在行设置/取消断点
3. 右击鼠标, 弹出菜单, 选择 设置/取消断点,
4. 主菜单 执行/设置取消断点, 也可以用 Alt-R / B 菜单快捷设置取消断点

执行 | 清除全部断点

清除程序中所有的断点。让程序全速执行

窗口 (W)

窗口 | 刷新




刷新打开的所有窗口, 及窗口里的数据。

窗口 | 项目窗口

打开项目窗口, 以便在项目中加入模块或包含文件。(图: 项目窗口)

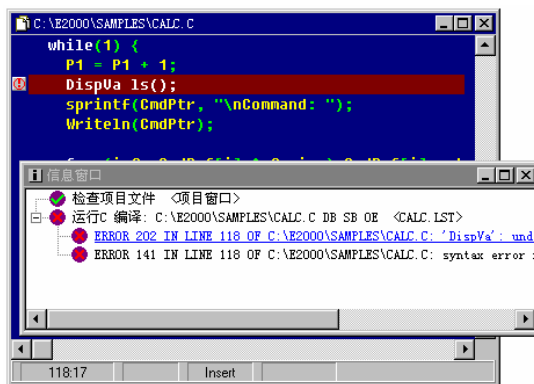
窗口 | 信息窗口

显示系统编译输出的信息。如果程序有错, 会以图标形式指出,

 表示错误,  表示警告,  表示通过

在编译信息行会有相关的生成文件,

双击鼠标左键, 或击右键在弹出菜单中选择“打开”功能, 可以打开相关文件。(如果有编译错误, 双击左键, 可以在源程序中指出错误所在行, 有时前一行或后一程序有错, 会造成当前行编译不通过. 而将错误定位在本行, 所以如果发现了错误, 但在本行没有发现错误, 可以查查本行上下的程序)。



例: 编译过程发现有错. 在信息窗口中看到在 CALC.C 文件第 118 行有 202 号错误,

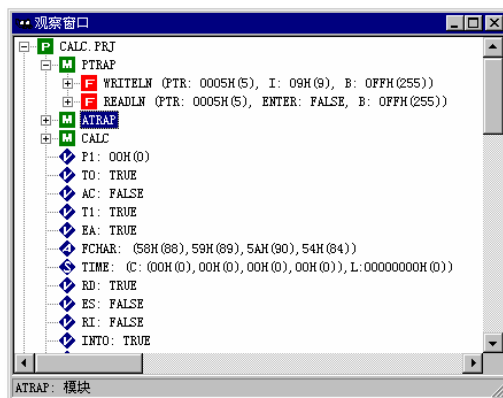
文字显示错误类型是, “ 'DispVa' undefined identifier” 即: 未定义 DispVa 标识符. 双击此信息行, 系统将打开 CALC.C 文件, 并且在源文件的 118 行, 指出有错, 可以看到, DispVa 和 ls() 中间有空格. 原来应为 DispVals().

窗口 | 观察窗口

项目编译正确后，可以在观察窗口中看到当前项目中的所有模块，及各模块中的所在过程和函数，及各个过程函数中的各个变量，结构。如果能充分利用观察窗口的强大功能，可以加快你开发速度。

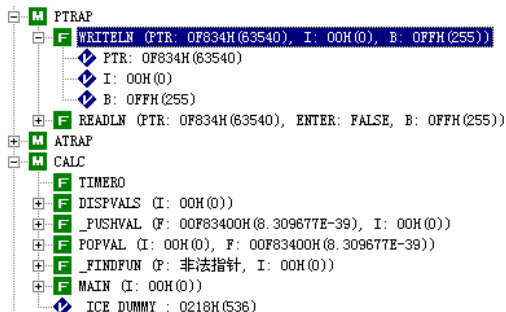
观察窗口也可以用观察数据时效分析，程序时效分析，代码覆盖以及影子存储器等分析功能的结果。（参见各分析功能使用）

- P** 表示当前项目，双击可以展开，观察到项目中的模块和项目所使用的变量
- M** 表示项目中所包括的模块，双击可以展开，观察到项目中包含的过程函数。



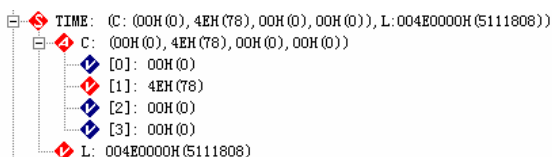
（图：观察窗口）

例：一个打开的项目，可以看项目中包括 PTRAP, ATRAP, CALC 三个模块，其中 PTRAP 展开，PTRAP 包括 WRITELN 和 READLN 两个函数，可以看到展开的 WRITELN 函数使用个三个简单变量：PTR, I, B。在展开的 CALC 模块中包含了六个函数。



- F** 表示模块中的函数，双击可以观察到模块中所用到的变量。
- V** 表示模块或函数中使用的简单变量。
- A** 表示模块或函数中使用的数组，双击可以展开数组，观察数组中各值的变化。
- S** 表示模块或函数中使用的结构，双击可以展开结构，观察结构内部变量值。
- P** 表示模块或函数中使用的指针
- L** 表示模块或函数中使用的标号

例：一个展开的结构。结构变量名为 TIME，它包括一个数组变量和一个长整型变量，通过展开的数组，可以看到数组有四个元素，其中第二个元素在上次执行过程中发



生了变化,长整形变量 L 也发生变化.

标记颜色为红,表示在上次执行过程中变量值发生变化.

标记颜色为橄榄色,表示在上次执行过程中,变量被访问过.

可以在弹出菜单中选择

[加入观察], 把当前行的变量放在窗口的最后,而不用展开复杂的项目,模块来观察某个变量.

[编辑观察项],可以打开“编辑观察窗口”,对当前变量进行修改,求值

[删除观察项],删除观察窗口后变量.不再对此变量进行观察.

[察看源程序],若窗口中当前行是模块文件,用此命令可以打开相应的模块文件.

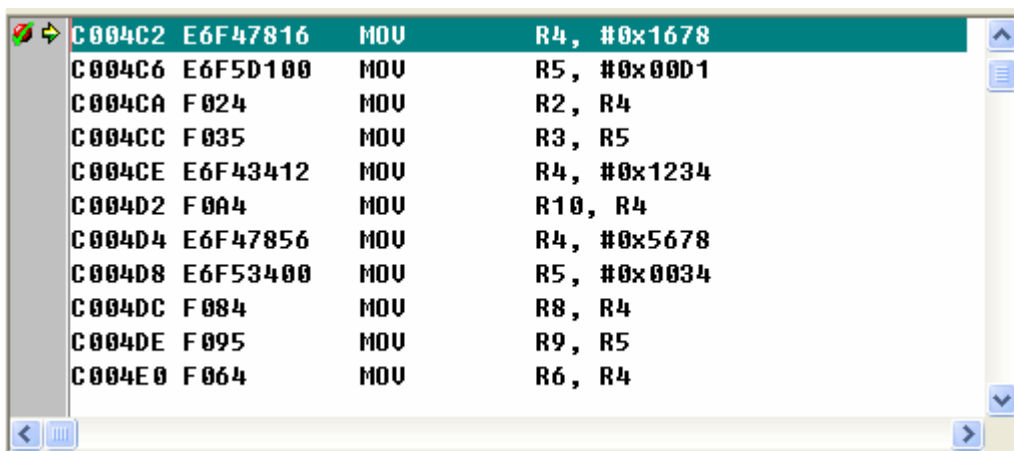
[展开],[收缩],用于展开/收缩当前行的函数,结构,数组.便于观察.

[修改], 对当前行的变量进行修改,在程序运行时,给出所要求的值.以观察程序在此值时,运行的结果.

[窗口总在最上面],若此项被选中,则窗口会一直显示在前面,覆盖其它窗口.建议选中此项,使窗口在最上面时,不要使窗口最大化,以免完全覆盖其它窗口.

在观察窗口最下面的状态行可以看到观察项的更详细信息,当你在点击一个项目时,可以看到该观察项的类型(模块,函数/过程),所在区域(CODE, DATA, BIT, XDATA 等到),地址,数据类型(unsigned char, unsigned int, real 等)

窗口 | CPU 窗口



C004C2	E6F47816	MOV	R4, #0x1678
C004C6	E6F5D100	MOV	R5, #0x00D1
C004CA	F024	MOV	R2, R4
C004CC	F035	MOV	R3, R5
C004CE	E6F43412	MOV	R4, #0x1234
C004D2	F0A4	MOV	R10, R4
C004D4	E6F47856	MOV	R4, #0x5678
C004D8	E6F53400	MOV	R5, #0x0034
C004DC	F084	MOV	R8, R4
C004DE	F095	MOV	R9, R5
C004E0	F064	MOV	R6, R4

通过 CPU 窗口,可以打开反汇编窗口, SFR 窗口和 REG 窗口。在反汇编窗口中可观察编译正确的机器码及反汇编程序,可以让你更清楚地了解程序执行过程。SFR 窗口中可以观察到单片机使用的 SFR(特殊功能寄存器)值和位变量的值。REG 窗口为 R0..R7、A、DPTR 等常用寄

寄存器的值。

反汇编窗口内为程序地址, 机器码, 反汇编码。在机器码窗口内也支持点屏功能, 在反汇编码处, 点击寄存器, 可以看到寄存器的值。

反汇编窗口的弹出菜单

执行到光标处：使程序从当前 PC 值, 全速执行到光标所在行, 用这种方法可以在调试程序时, 跳过一些不必要的指令, 将程序停到所要求的位置上。

转到指定地址/标号：将光标跳到某个地址或标号所在位置, 以便察看相应的程序, 或使用“执行到光标处”功能, 也可以设置断点, , 将程序全速执行到相应位置。

转到当前 PC 所在行：将光标跳到 PC 所在行, 由于在检查程序时, 可能会将 PC 所在行移出当前窗口, 用这种方法可找回 PC 所在行。

取消/设置断点：在光标所在行, 设置断点, 使程序全速执行到此处。若此行已是断点, 再次点击将取消该断点。

寄存器窗口的弹出菜单

加入观察：将当前寄存器放入观察窗口, 以方便随时察看。

修改：修改当前寄存器值。在程序执行时, 可以用这种方法, 把寄存器值改为你所指定的值, 从而观察程序在此值时运行的结果。

窗口 | 数据窗口



数据窗口中可以看到存贮器的数据值，红色的为上一步执行过程中，改变过的值，窗口状态栏中为选中数据的地址，可以在选中的数据上直接修改数据的十六进制值，也可以用弹出菜单的修改功能，修改选中的数据值。

弹出菜单：

修改：修改选中数据的值，可以输入十进制，十六进制，二进制的值，与直接修改不同的是，用这种方法可以输入多种格式数据，而直接修改只能输入十六进制数据。46（十进制），2EH（十六进制），00101110B（二进制）都是有效的数据格式，

转到指定地址/标号：将数据地址直接转到指定的地址和标号所在的位置。

生成数据源码：将窗口中某段数据转换成源程序方式的数据，可以贴到你的源程序中。

块操作：对窗口中的数据块进行填充、移动、写文件、读入等操作。

显示为：选择不同的数据类型显示数据内容，可以是字节方式（BYTE），也可以是字方式（WORD，两字节），可以是长整型（LONGINT，四字节），也可以是实数型（REAL，四字节）。这里是选择整个窗口的显示方式，如果想指定个别数据的显示方式，可以用主菜单[执行|编辑观察项]功能，选择所要选择的显示类型。（参见编辑观察项窗口）

显示列数：将窗口中数据以 4 列、8 列、16 列方式显示。适应不同需要。

程序数据窗口显示的是编译后程序码，状态栏显示的是选中数据的地址，可以对在选中数据上直接修改程序数据的十六进制值，也可以对程序数据进行‘块填充’，‘块移动’操作，也可以读入一段二进制代码插入程序数据中，也可以将程序数据中的某段代码写文件中。

弹出菜单

修改：修改选中数据的值，可以输入十进制，十六进制，二进制的值，与直接修改不同的是，用这种方法可以输入多种格式数据，而直接修改只能输入十六进制数据。46（十进制），2EH（十六进制），00101110B（二进制）都是有效的数据格式。

生成数据源码：将窗口中某段数据转换成源程序方式的数据，可以贴到你的源程序中。

块操作：对程序数据以块的方式进行操作。在窗口中按往左键拖动，可以选择块。

块填充：将选中的块内数据值，填充为指定值。

块移动：将选中的块移动到指定地址。

读文件：读入二进制代码文件，插入的指定的地址内。（参见‘调入目标文件’功能）

写文件：将程序数据指定地址的一段代码写入文件。（参见‘保存目标文件’功能）

显示为：选择不同的数据类型显示数据内容，可以是字节方式（BYTE），也可以是字方式（WORD，两字节），可以是长整型（LONGINT，四字节），也可以是实数型（REAL，四字节）。这里是选择整个窗口的显示方式。

窗口 | 断点窗口

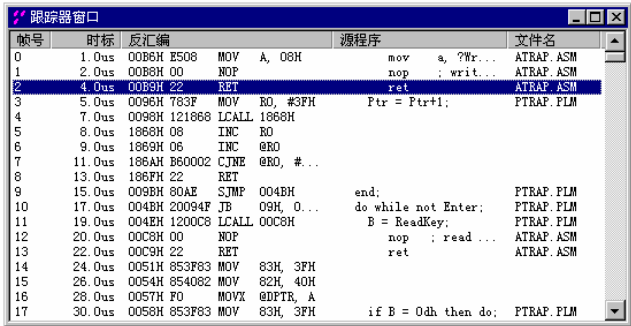
通过断点窗口可以管理项目内的断点。可以在断点窗口中直观地看到断点的行号，内容，可以通过断点迅速定位程序所在的位置。

窗口 | 书签窗口

通过书签窗口可以管理项目内的书签，在项目中迅速定位程序位置。

窗口 | 跟踪窗口

显示跟踪器捕捉到的程序执行的轨迹，其中可以看到帧号，时标，反汇编程序，对应的源程序和程序所在的文件名。



帧号	时标	反汇编	源程序	文件名
0	1.0us	00B6H E508 MOV A, 08H	mov a, ?Wr...	ATRAP.ASM
1	2.0us	00B8H 00 NOP	nop ; writ...	ATRAP.ASM
2	4.0us	00B9H 22 RET	ret	ATRAP.ASM
3	5.0us	0096H 783F MOV R0, #3FH	Ptr = Ptr+1;	PTRAP.PLM
4	7.0us	0096H 121868 LCALL 1868H		
5	8.0us	1868H 08 INC R0		
6	9.0us	1869H 06 INC @R0		
7	11.0us	186AH B60002 CJNE @R0, #...		
8	13.0us	186FH 22 RET		
9	15.0us	009BH 80AE SJMP 004BH	end;	PTRAP.PLM
10	17.0us	004BH 20094F JB 09H, 0...	do while not Enter;	PTRAP.PLM
11	19.0us	004BH 1200C8 LCALL 00C8H	B = ReadKey;	PTRAP.PLM
12	20.0us	00C8H 00 NOP	nop ; read ...	ATRAP.ASM
13	22.0us	00C9H 22 RET	ret	ATRAP.ASM
14	24.0us	0051H 853F83 MOV 83H, 3FH		
15	26.0us	0054H 854082 MOV 82H, 40H		
16	28.0us	0057H F0 MOVX @DPTR, A		
17	30.0us	0058H 853F83 MOV 83H, 3FH	if B = 0dh then do;	PTRAP.PLM

窗口 | 工具条

通过工具条，可以打开/关闭菜单上的各功能的快捷按钮。

窗口 | 排列窗口

对打开的程序窗口进行管理。可叠排、竖排、横排、最小化源程序窗口。

仿真器 (0)

仿真器 | 仿真器设置 | 语言设置

设置项目编译语言的路径，命令行选项。

仿真器设置

编译器 | 目标文件 | 仿真器

编译器路径

C:\KEIL\C166\

浏览...

编译器选择

☐ GUN C166 编译器

☒ KEIL C166 编译器

☐ Tasking C166 编译器

编译器设置

缺省显示格式

☐ 二进制

☐ 十进制

☐ 十六进制

☐ 字符

☒ 混合十, 十六进制

好

取消

帮助

仿真器设置

语言 | 目标文件 | 仿真器 | 通信设置

编译器路径

C:\KEIL\C51\

ASM 命令行

DB SB

C 命令行

DB SB OE

PL/M 命令行

DB SB XR ROM (LARGE)

LINK 命令行

RS (256)

编译器选择

☐ 伟福汇编器

☐ Keil C (V4或更低), 英特尔PL/M51, 英特尔汇编器

☒ Keil C (V5或更高), 英特尔PL/M51, 英特尔汇编器

缺省显示格式

☐ 二进制

☐ 十进制

☐ 十六进制

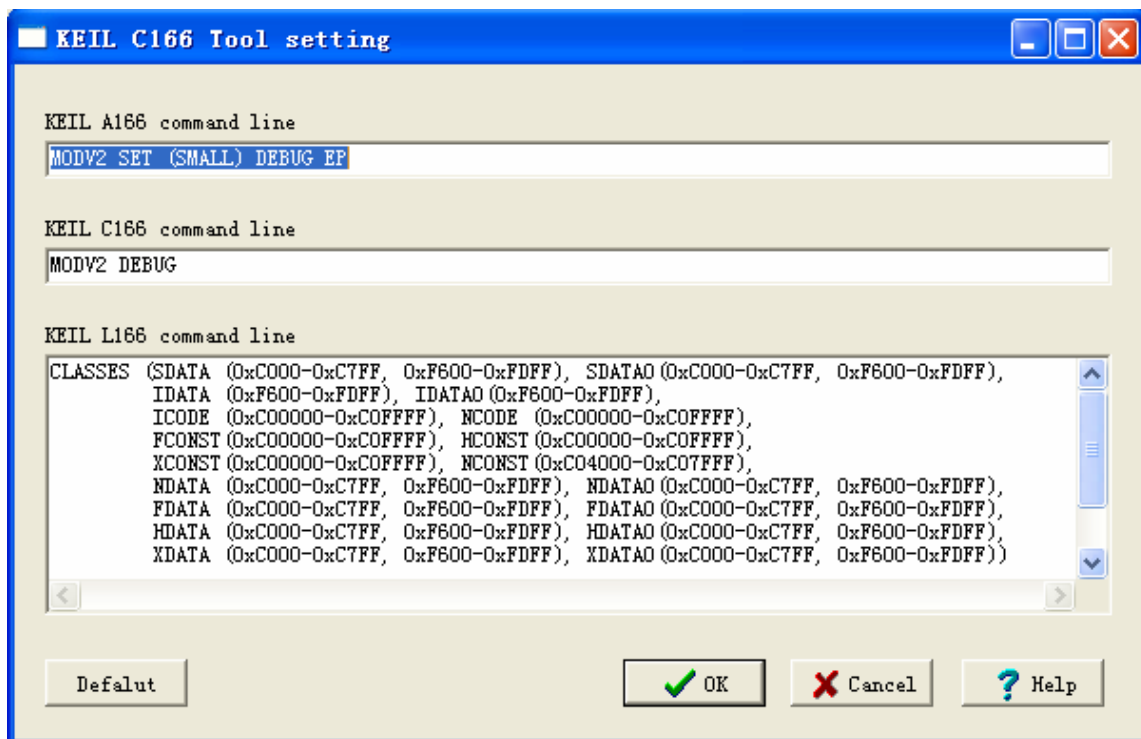
☐ 字符

☒ 混合十, 十六进

好

取消

帮助



[编译器路径]:指明本系统汇编器,编译器所在位置,系统缺省编译器在 C:\KEIL\C166 文件夹下. 本系统使用的编译器为第三方软件, 你应从其它途径获得.

[A166 命令行]: 汇编命令行

[C166 命令行]: C 编译命令行

[L166 命令行]: 连接命令行

注: 除非你对命令行参数非常了解, 并且确实需要修改这些参数, 一般情况下, 不需要修改系统给出的缺省参数. 以免系统不能正常编译.

详情的命令行说明请参考 KEIL 手册

[编译器选择]: 现只支持 KEIL.

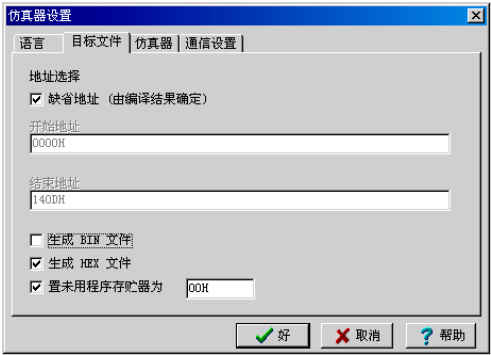
[缺省显示格式]:

指定观察变量显示的方式, 一般为混合十/十六进制.

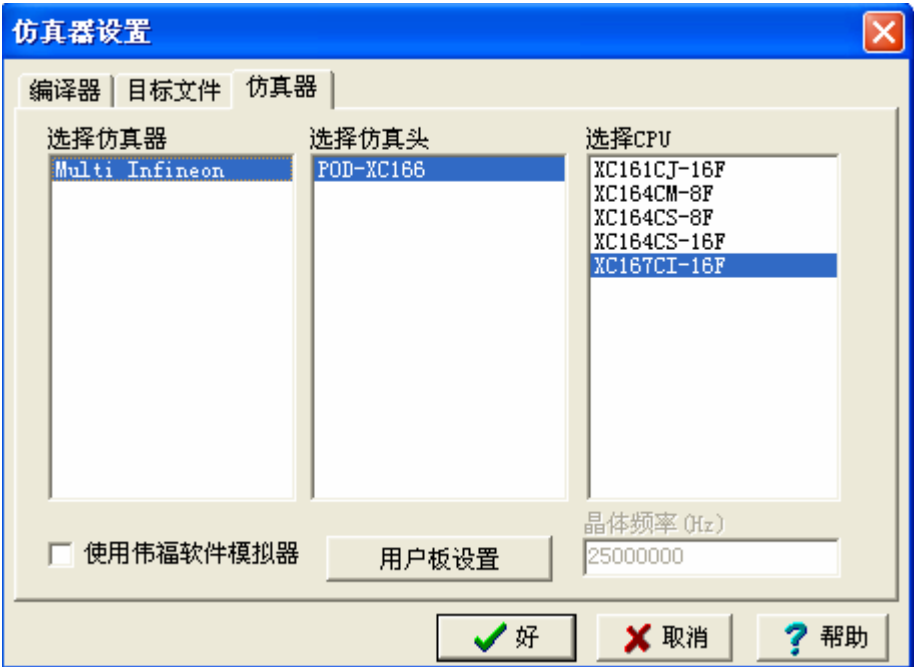
目标文件设置

设置生成的目标文件的地址，及生成目标文件的格式。

一般情况下，地址选择为缺省方式，即由编译确定。如果你想重新定位你的程序就要指定地址，方法是：去掉[缺省地址]前面的选择。在开始地址，结束地址处填入相应的地址。编译可以生成 BIN(二进制)格式和 HEX(英特尔)格式的目标文件，可以根据你的需要，选择相应的格式。



仿真器设置



选择所使用的仿真器类型，仿真头类型，以及所仿真 CPU 的类型。如果使用硬件仿真，请去掉“使用伟福软件仿真”前的选择，在仿真头设置中可以设置该仿真头的特殊功能。 [选择仿真器]:框内为本系统所支持的仿真器类型. 选择正确仿真器.

[选择仿真头]: 框内为相应仿真器能支持的仿真头类型, 选择所使用的仿真头.

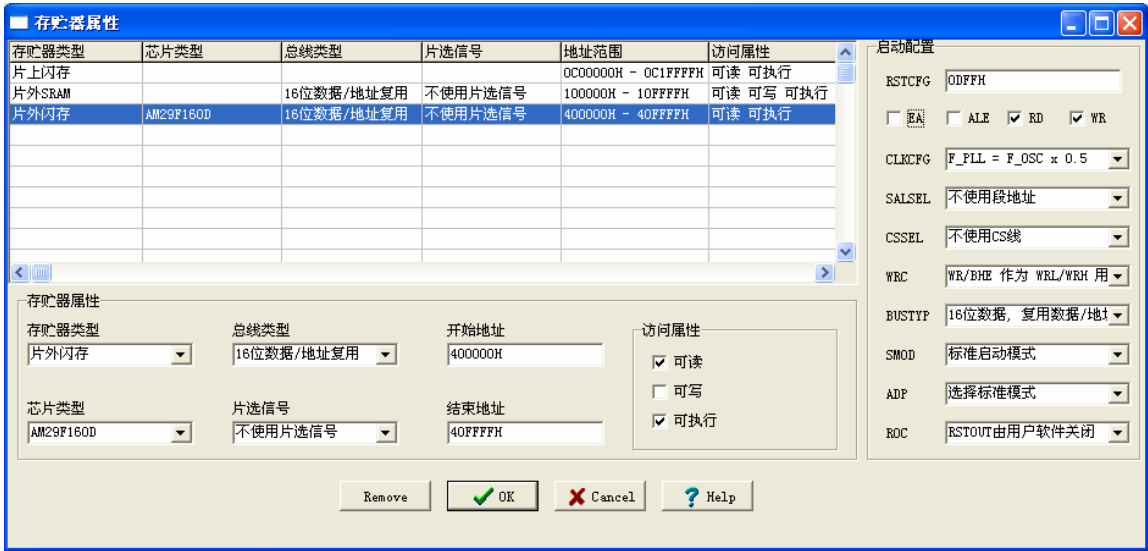
[选择 CPU]: 框内为选择的仿真器和仿真头等能进行仿真支持的 CPU.

[使用伟福软件模拟器]:使用伟福软件模拟器, 可以在完全脱离硬件仿真器情况下, 对软件进行模拟执行. 如果使用硬件仿真器, 请不要选择使用伟福软件模拟.

[晶体频率]: 在使用伟福软件模拟功能时, 用来计算在软件模拟环境下程序执行时间。在外

设中串行口的波特率也是依据此频率计算出的。

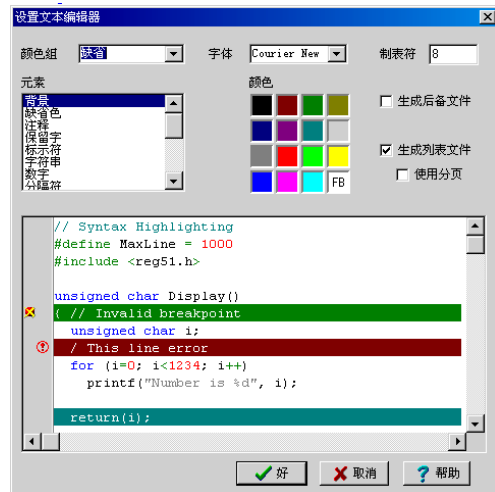
[用户板设置]：可以设置用户板的存贮空间分配及启动配置
在仿真前必须对存贮器配置



仿真器 | 设置文本编辑器

在设置文本编辑器的对话框中，你可以设置你自己喜爱的文本编辑环境。

文本编辑器设置对话框



帮助(H)

帮助 | 关于

帮助 | CHINESE

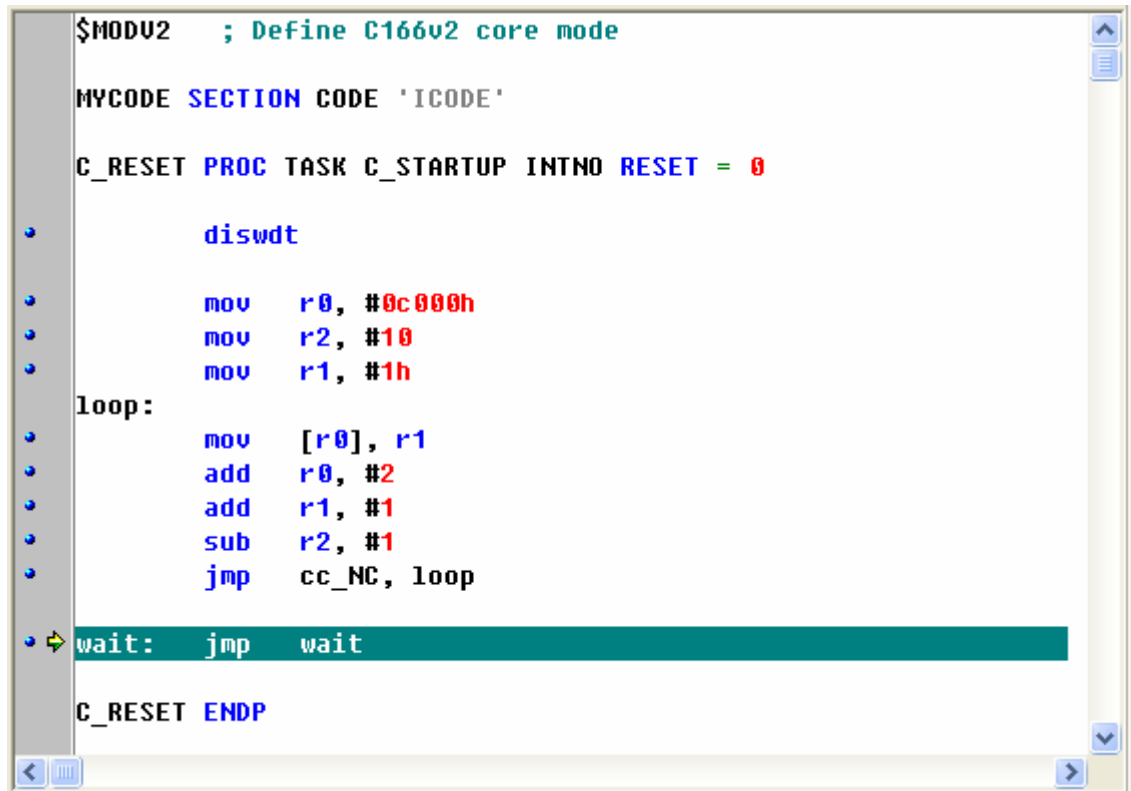
选择中文或英文显示方式，适应不同操作系统的需要。

快速入门

1. 建立你的新程序

选择菜单[文件 | 新建文件]功能

出现一个文件名为 NONAME1 的源程序窗口，在此窗口中输入以下程序



```
$MODV2      ; Define C166v2 core mode

MYCODE SECTION CODE 'ICODE'

C_RESET PROC TASK C_STARTUP INTNO RESET = 0

    diswdt

    mov     r0, #0c000h
    mov     r2, #10
    mov     r1, #1h

loop:
    mov     [r0], r1
    add     r0, #2
    add     r1, #1
    sub     r2, #1
    jmp     cc_NC, loop

wait:  jmp     wait

C_RESET ENDP
```

输入程序后的窗口如图，现在要做的是将此文件存盘。

2. 保存你的程序

选择菜单[文件 | 保存文件]或[文件 | 另存为]功能

给出文件所要保存的位置，例如：C:\XC166\SAMPLES 文件夹，再给出文件名 MY1.ASM。保存文件。文件保存后，程序窗口上文件名变成了：

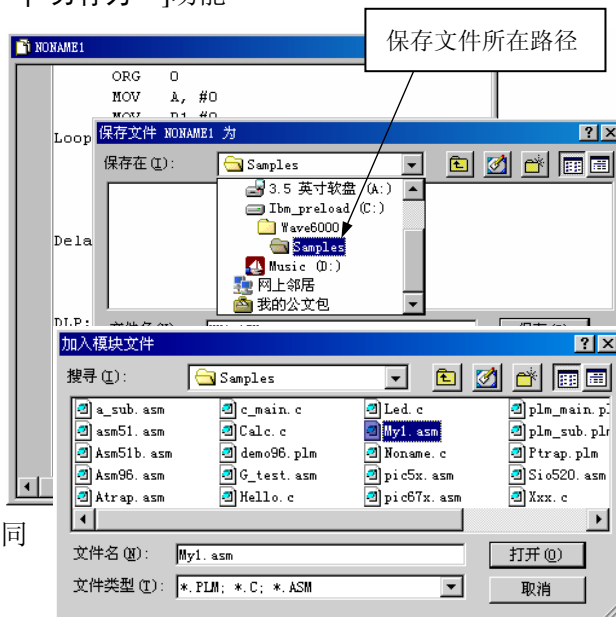
C:\WACE166\SAMPLES\MY1.ASM

3. 建立新的项目

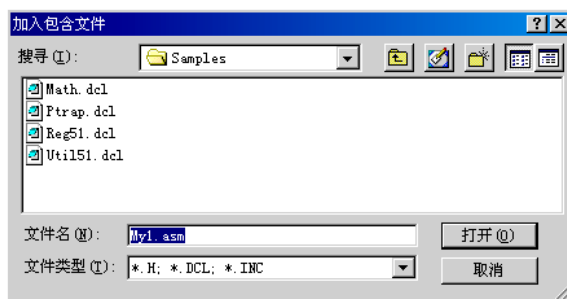
选择菜单[文件 | 新建项目]功能

新建项目会自动分三步走。

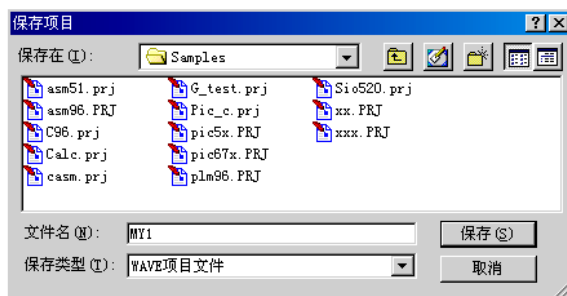
A) 加入模块文件。在加入模块文件的对话框中选择刚才保存的文件 MY1.A66，按打开键。如果你是多模块项目，可以同时选择多个文件再打开。



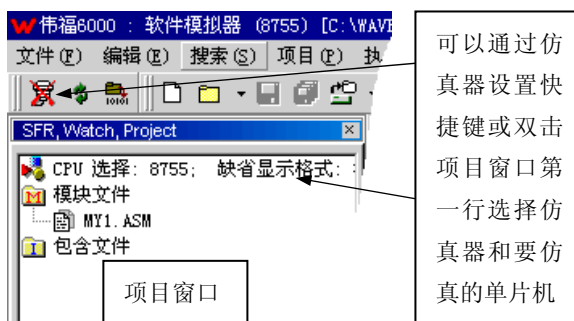
B) 加入包含文件。在加入包含文件对话框中，选择所要加入的包含文件（可多选）。如果没有包含文件，按取消键。



C) 保存项目。在保存项目对话框中输入项目名称。MY1 无须加后缀。软件会自动将后缀设为“.PRJ”。按保存键将项目存在与你的源程序相同的文件夹下。



项目保存好后，如果项目是打开的，可以看到项目中的“模块文件”已有一个模块“MY1.A66”，如果项目窗口没有打开，可以选择菜单[窗口 | 项目窗口]功能来打开。可以通过仿真器设置快捷键或双击项目窗口第一行选择仿真器和要仿真的单片机



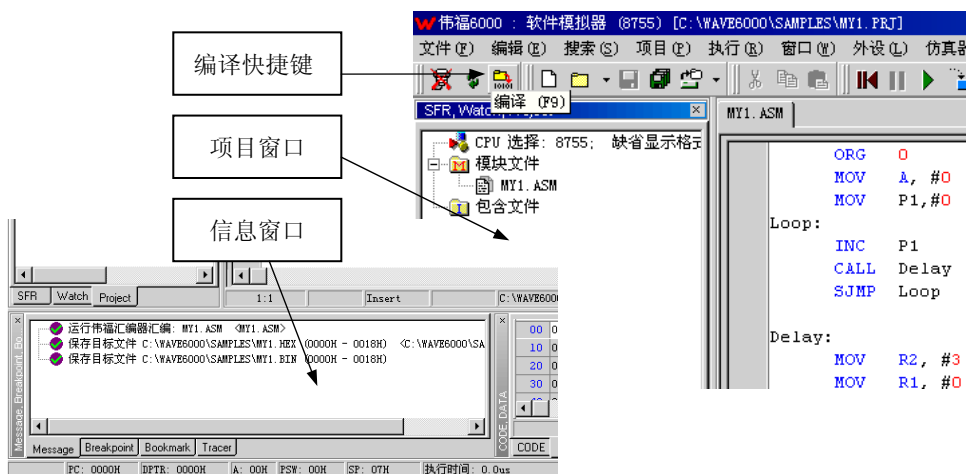
4. 设置项目

选择菜单[设置 | 仿真器设置]功能或按“仿真器设置”快捷图标或双击项目窗口的第一行来打开“仿真器设置”对话框

在“仿真器”栏中，选择仿真器类型和配置的仿真头以及所要仿真的单片机。

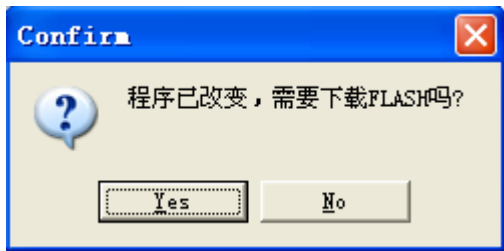
5. 编译你的程序

选择菜单[项目 | 编译]功能或按编译快捷图标或 F9 键，编译你的项目。



在编译过程中，如果有错可以在信息窗口中显示出来，双击错误信息，可以在源程序中定位所在行。纠正错误后，再次编译直到没有错误。在编译之前，软件会自动将项目和程序存盘。在编译没有错误后，就可调试程序了，首先我们来单步跟踪调试程序。

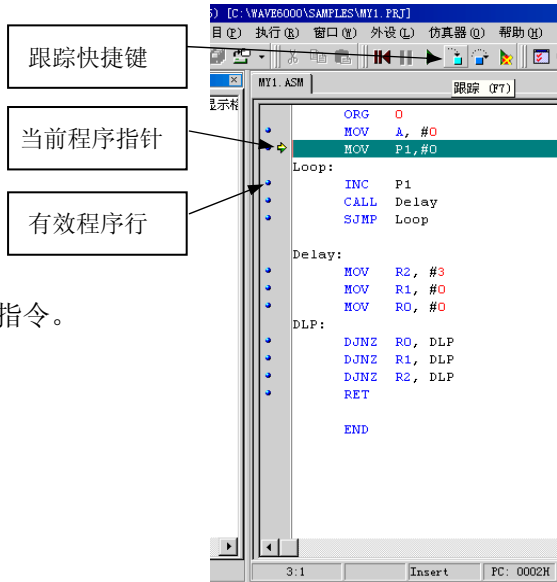
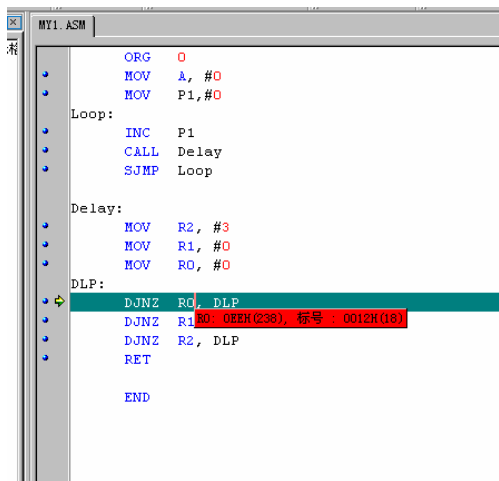
如果程序在 FLASH 中，系统会提问你是否要写入新代码，按[YES]。



6. 单步调试程序

选择[执行 | 跟踪]功能或按跟踪快捷图标或按 F7 键进行单步跟踪调试程序

单步跟踪就一条指令一条指令地执行程序，若有子程序调用，也会跟踪到子程序中去。你可以观察程序每步执行的结果，“=>”所指的就是下次将要执行的程序指令。由于条件编译或高级语言优化的原因，不是所有的源程序都能产生机器指令。源程序窗口最左边的“o”代表此行为有效程序，此行产生了可以指行的机器指令。



我们有“执行到光标处”的功能，将光标移到程序想要暂停的地方，本例中为延时子程序返回后的“jmp wait”行。选

择菜单[执行 | 执行到光标处]功能或 F4 键或弹出菜单的“执行到光标处”功能。程序全速执行到光标所在行。

将光标移到源程序窗口的左边灰色区，光标变成“手指圈”，单击左键设置断点，也可以用弹出菜单的“设



置/取消断点”功能或用 **Ctrl+F8** 组合键设置断点。如果断点有效图标为“红圆绿勾”，无效断点的图标为“红圆黄叉”。断点设置好后，就可以用全速执行的功能，全速执行程序，当程序执行到断点时，会暂停下来，这时你可以观察程序中各变量的值。

7. 连接硬件仿真

按照说明书，将仿真器通过并行电缆连接计算机上，将仿真头接到用户板，检查接线是否有误，确信没有接错后，接上电源，打开仿真器的电源开关。

参见第 4 步，设置项目，在“仿真器”和“通信设置”栏的下方有“使用伟福软件模拟器”的选择项。将其前面框内的勾去掉。在通信设置中选择正确的串行口。按“好”确认。

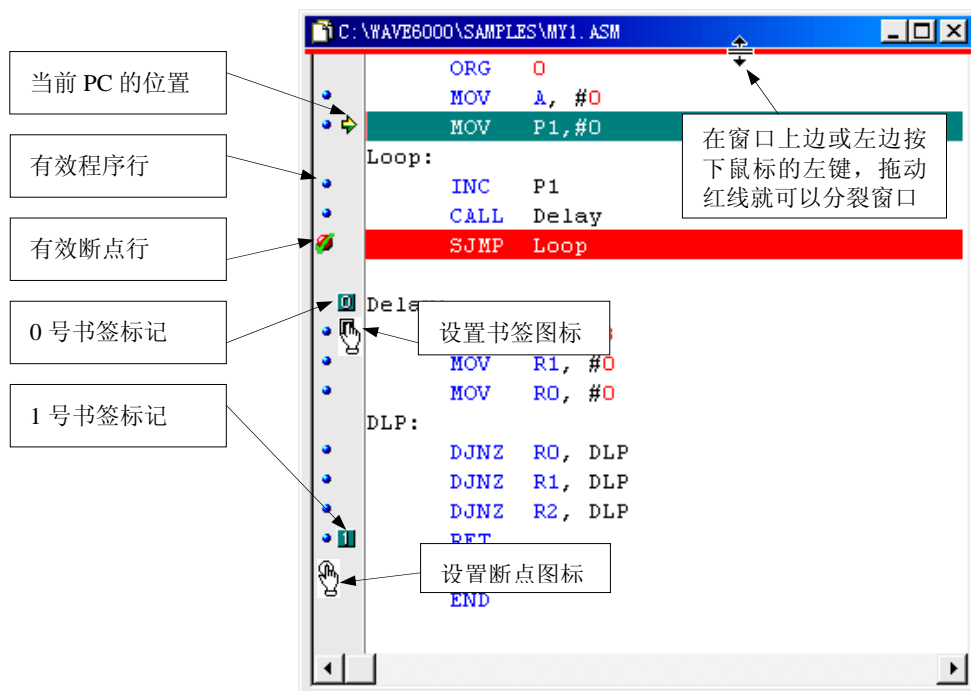
在 XC866 选择 **Multi-Infineon** 仿真器，使用 **JTAG0**。

如果仿真器初始化过程中有错，软件就会再次出现仿真器设置对话框，这时你应检查仿真器、仿真器的选择是否有错，硬件接线是否有错，检查纠正错误后，再次确认。直至显示如图的硬件仿真确认对话框。

到此为止，你已经学会使用伟福的仿真环境了。在使用过程中，你会逐步提高自己的技能。伟福仿真器的更多功能可参考本说明书的其它部分。

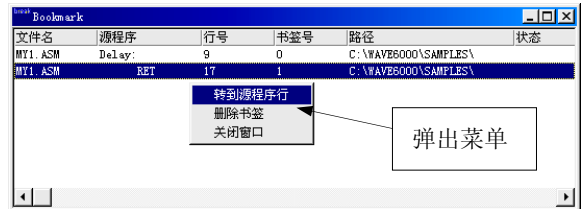
伟福文本编辑器的使用

伟福文本编辑器用来输入程序，使用方便。具有与 C 语言、汇编语言、PLM 语言语法相关的彩色显示，使编写程序更加轻松，观察程序醒目。并且用户可按照自己的喜好自己设置颜色，享受个性化编程带来的乐趣。可以在编辑窗口中设置断点、书签，用于快速定位程序，对于编写、分析、比较、检查较长的、复杂的程序非常有帮助。查找功能可以在程序中查找、替换字符串。在编辑窗口中，可以查找配对符号，如找到 ‘{’ 相对的 ‘}’ 或找到与 ‘(’ 相对 ‘)’，并且将中间的部分加亮显示，这样在复杂的嵌套中确定程序的块结构。可以在编辑窗口中对多行程序同进同退，帮助您编写优美、整洁的程序。窗口分隔功能可将源程序窗口分成两个或三个完全独立的编辑窗口，而所编辑的内容却是同一程序，为分析、比较检查大程序提供方便。



设置断点、书签

编辑窗口的左边界用于显示断点、书签。将鼠标移到边界的右半边，光标变成“手及方块”此时单击鼠标左键就可以设置书签，共可以设置多达 10 个书签，标号从 0 到 9，如果当前位置上已有书签，单击鼠标则去除此书签，书签的添加、删除操作也可以在书签窗口中实现。在书签窗口中，双击书签号就可以将编辑窗口中的光标快速定位到书签所在的行。此功能对于在大程序迅速定位很有帮助。移到左半边则可以设置断点。

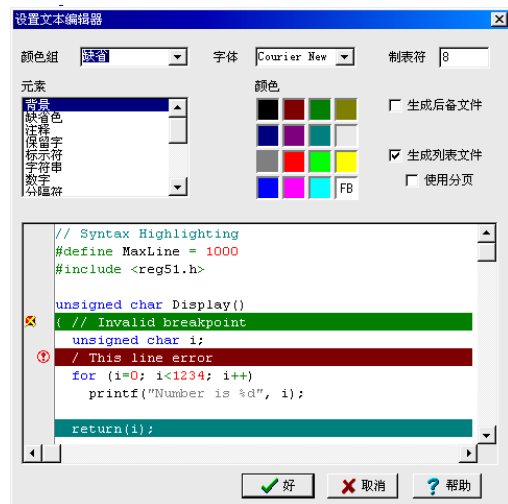


书签窗口

将光标移到编辑窗口边界的左半边，光标变成“手及圈”，单击鼠标左键设置断点，也可以用弹出菜单的“设置/取消断点”功能或用 **Ctrl+F8** 组合键设置断点。如果不在调试程序，断点图标为“红圆”，在调试程序时，如果断点有效图标为“红圆绿勾”，若当前行已有断点，单击左键就会删除此断点。同样断点的添加、删除也可以在断点窗口中操作。断点除了在调试时让程序暂停的功能外，断点也可以象书签一样，可以快速定位程序的位置。

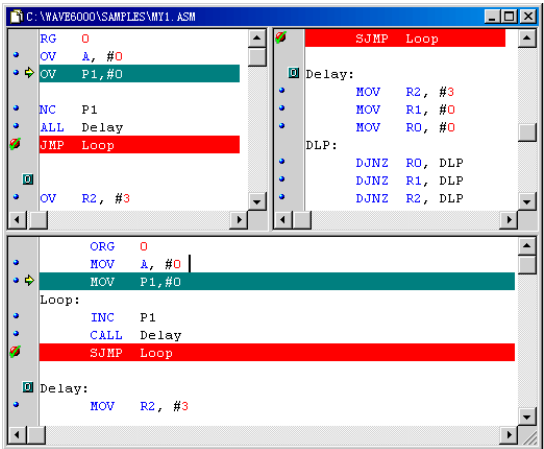
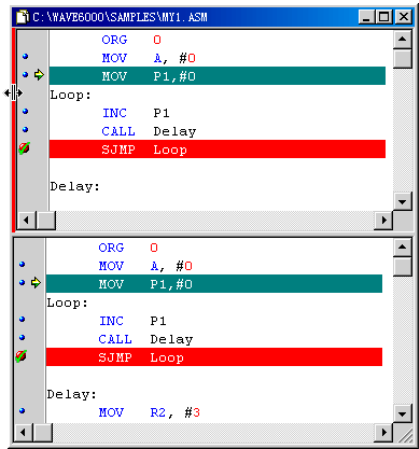
设置个性化编辑器颜色

选择主菜单的[仿真器 | 设置文本编辑器]功能，就可对编辑器的颜色进行设置。软件环境中已经预定义了四组颜色，为“缺省”、“古典”、“高亮”、“海洋”。用户如果对这四种都不满意，可以自己定义自己喜爱的颜色，在“元素”栏中选择要定义的元素，在“颜色”栏中定义背景色和前景色。在“字体”栏中选择字体。定义颜色和字体后，可以下面的图例中看到编辑器产生的效果。



分裂多窗口

源程序编辑窗口可以分割成两个或三个窗口，用于观察同一程序的不同位置。各个分窗口的横竖滚动棒可以独立控制。在编辑窗口的上方按下鼠标左键，就会出现一条红线，表示窗口分割线，拖动红线大于一定距离松开，就可以分裂窗口。若想关闭分窗口，在窗口分界线上按下鼠标左键，也会出现红线，拖动到上/下边小于一定距离松开，就会关闭分窗口。若相再分出一个窗口，可在窗口左边上方按下鼠标左键（如左图），拖动红线可分出第三个窗口（如右图）。

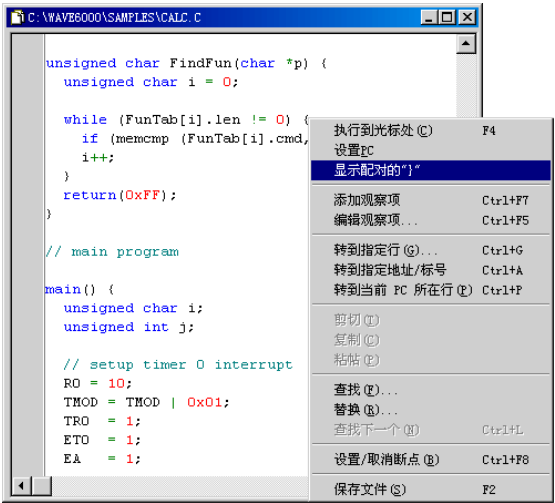


查找匹配符号

在编写多重嵌套结构的程序时，有时非常想知道某一符号对应的符号在何处，这用到了查找匹配符号的功能。当光标放在要查找的符号上，按鼠标右键弹出菜单，选择“查找匹配符号”功能，两配对符号之间的程序会被加亮显示

多行文本同进同退

用多行文本同进同退功能，可以将程序排列整齐，首先选中需要排列的程序块，按住 Ctrl 键，按“U”键可将文本左移一位，若按“I”键会将文本右移一位。



1 简介

本手册介绍 Infineon XC164CS 用户板

XC164CS 基于 0.25 μ C166S V2 内核，性能大于二倍的标准 C166 内核。

XC164 用户板含有 FLASH、SRAM、晶振、电源及外设驱动电路。

2 XC164 用户板功能

2.1 功能简介

- Infineon XC164 微控制器 (100TQFP 封装)
- 64K 16 位 SRAM
- 512K 16 位 FLASH
- 板上集成有 2.5V DC-DC
- 晶振插座 (装有 8M 晶振)
- 板上有二个 CAN 驱动、二个 ASC (RS232) 串口驱动、一个 JTAG 口 (用于仿真)
- 四个 LED 发光二极管、四个按键输入

连接器

SUB-D9 串口连接器

10 脚双排针 (2x5) 串口连接器

双个 10 脚双排针 (2x5) CAN 总线连接器

四个 26 脚双排针 (2x13) XC166 管脚连接器

其它器件

二个电源指示灯 (5V 及 2.5V)

RSTIN/RSTOUT 指示灯

复位按键

KEY EN 按键允许跳线器

CAN 允许跳线器

CAN 管脚选择跳线器

2.2 DIP 开关选择器

XC164 用户板由四个 DIP 开关选择器见表一

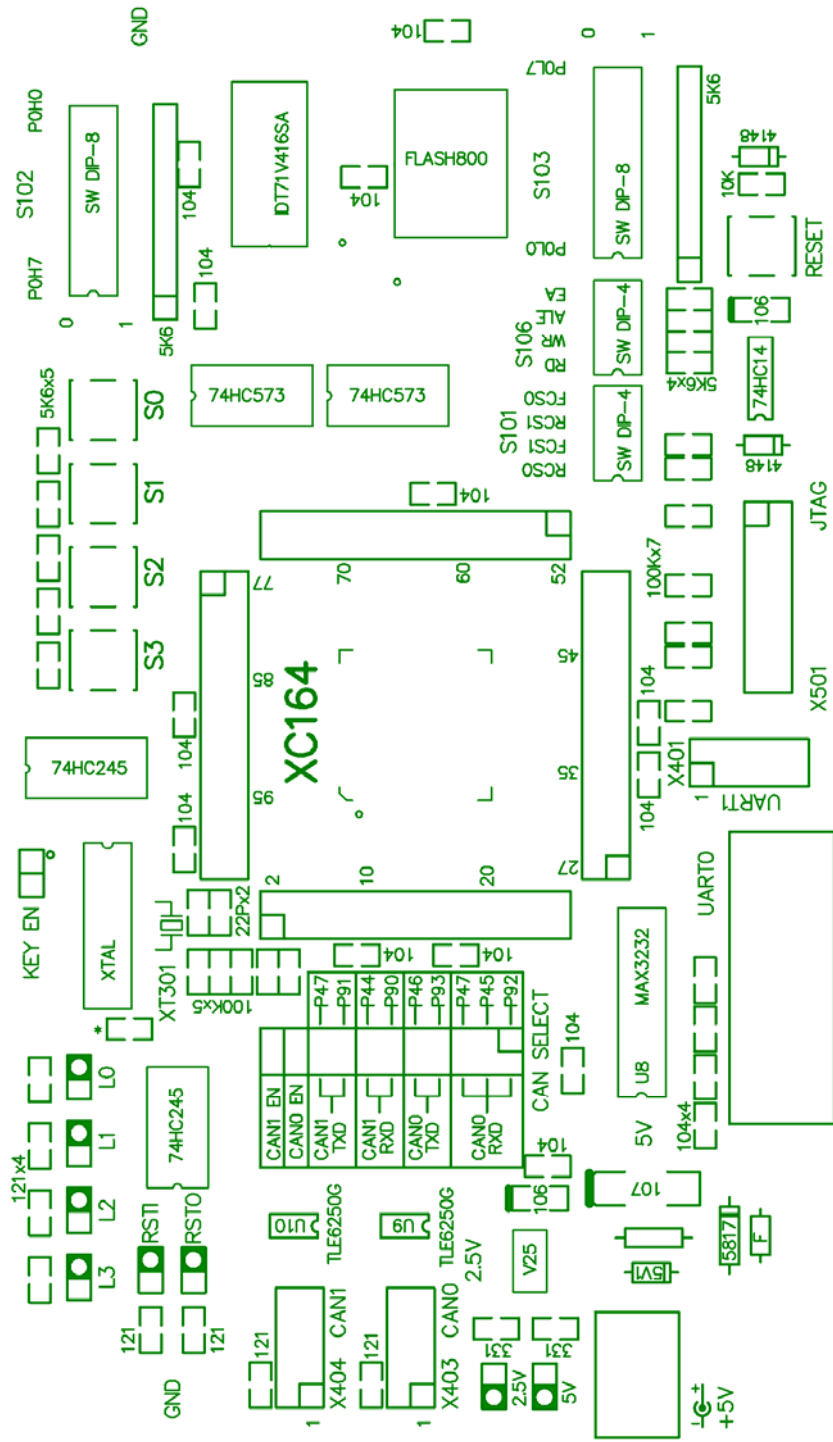
表一 DIP 开关选择器

	开关	功能
1	S102	P0 高位配置
2	S103	P0 低位配置
3	S101	CS 片选配置
4	S106	Startup 启动配置

通过这些 DIP 开关，你可以配置：

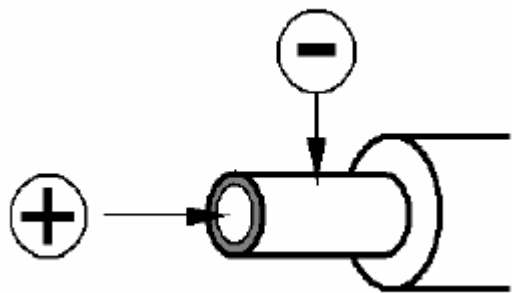
- S102、S103 启动配置
- RSTOUT 管脚配置
- 启动模式(内部启动、外部启动、启动装载等)
- 外部总线方式(8 位或 16 位、地址/数据脚复用、地址/数据脚独立)
- WR 写信号工作方式
- CS 片选线个数
- 段地址线个数
- 输入频率范围
- PLL/OWD 功能打开/关闭
- CS0/CS1 与外部 SRAM 及 FLASH 的连接

2.3 用户板布局



2.4 供电

XC164 用户板使用单 5V 直流电源，电源插座内正外负，如图：



3. 缺省配置

XC164 用户板的缺省配置：

- 程序从内部 FLASH C00000H 开始执行
- XC164 使用 16 位地址/数据复用方式
- 使用 2 根 CS 脚，2 个段地址线
- 晶振配置为 $f = f_{osc} \times 0.5$ (8MHz 输入)，CPU 工作在 4M
- CS0 不接，SRAM 不工作
- CS1 不接，FLASH 不工作
- EA 脚接高

4 跳线器及 DIP 开关

名称	缺省值	配置
S102	P0 高位配置	1B ON=0 OFF=1
S103	P0 低位配置	FF ON=0 OFF=1
S101	CS 片选配置	全 OFF 不接 SRAM 及片外 FLASH
S106	启动配置	全 OFF EA=1, RD=1, WR=1, ALE=0

KEY_EN	不接	四个按钮不连接
CAN1_EN	不接	CAN1 不连接
CAN0_EN	不接	CAN0 不连接
CAN1_TXD	不接	选择 CAN1_TXD 引脚
CAN1_RXD	不接	选择 CAN1_RXD 引脚
CAN0_TXD	不接	选择 CAN0_TXD 引脚
CAN0_RXD	不接	选择 CAN0_RXD 引脚

4.1 连接器

4.1.1 RS232 连接器 ASC0

4.1.3 ASC1 连接器

4.1.4 CAN0 连接器

4.1.5 CAN1 连接器

4.1.6 JTAG 连接器

4.2 DIP 开关配置

4.2.1 S106

DIP 开关 S106 用于配置启动

	RD	WR	ALE	EA
ON	0	0	1	0
OFF	1	1	0	1

常用的二种配置是:

RD	WR	ALE	EA	启动地址
OFF (1)	OFF (1)	OFF (0)	OFF (1)	片内 C00000
OFF (1)	OFF (1)	OFF (0)	ON (0)	片外 000000

4.2.2 S101

DIP 开关 S101 用于配置 CS0 及 CS1

	RCS0	FCS1	RCS1	FCS0
ON	SRAM 接 CS0	FLASH 接 CS1	SRAM 接 CS1	FLASH 接 CS0
OFF	SRAM 不接 CS0	FLASH 不接 CS1	SRAM 不接 CS1	FLASH 不接 CS0

常用的二种配置是:

RCS0	FCS1	RCS1	FCS0	配置
ON	OFF	OFF	OFF	SRAM 接 CS0, FLASH 不接
OFF	OFF	OFF	ON	FLASH 接 CS0, SRAM 不接

4.2.3 S102 及 S103

DIP 开关 S102 及 S103 用于配置外部启动, 缺省值为 1BFF

5 例子程序

在安装目录的 SAMPLES 目录下有 7 个例子程序，它们是：

EX1A.PRJ、EX1AR.PRJ、EX1C.PRJ、EX2A.PRJ、EX2C.PRJ、EX3C.PRJ、EX4C.PRJ

EX1A.PRJ

发光管显示程序，用汇编语言编写。

程序在片内 FLASH 中，要求 S106、S101 配置为：

S106 RD=OFF(1) WR=OFF(1) ALE=OFF(0) EA=OFF(1)

S101 RCS0=OFF FCS1=OFF RCS1=OFF FCS0=OFF

EX1AR.PRJ

发光管显示程序，用汇编语言编写。

程序在片外 SRAM 中，要求 S106、S101、S102、S103 配置为：

S106 RD=OFF(1) WR=OFF(1) ALE=OFF(0) EA=ON(0)

S101 RCS0=ON FCS1=OFF RCS1=OFF FCS0=OFF

S102 及 S103 = 1BFF

这个程序同 EX1A，只是程序放在片外的 SRAM 中

EX1C.PRJ

发光管显示程序，用 C 语言编写。

程序在片内 FLASH 中，要求 S106、S101 配置为：

S106 RD=OFF(1) WR=OFF(1) ALE=OFF(0) EA=OFF(1)

S101 RCS0=OFF FCS1=OFF RCS1=OFF FCS0=OFF

这个程序同 EX1A，只是程序用 C 语言编写

EX2A.PRJ

发光管显示及按钮读入程序，用汇编语言编写。

程序在片内 FLASH 中，要求 S106、S101 配置为：

S106 RD=OFF(1) WR=OFF(1) ALE=OFF(0) EA=OFF(1)

S101 RCS0=OFF FCS1=OFF RCS1=OFF FCS0=OFF

KEY EN 短接 使用 4 个按钮

EX2C.PRJ

这个程序同 EX2A，只是程序用 C 语言编写

EX3C. PRJ

串口通信程序

程序在片内 FLASH 中，要求 S106、S101 配置为：

S106 RD=OFF(1) WR=OFF(1) ALE=OFF(0) EA=OFF(1)

S101 RCS0=OFF FCS1=OFF RCS1=OFF FCS0=OFF

在 PC 机上运行超级终端程序，设置为 2400 波特率，8 个数据位，1 个停止位
这个例子只是简单的将 PC 机传来数据回传

EX4C. PRJ

外部 FLASH 编程程序

程序在片内 FLASH 中，要求 S106、S101 配置为：

S106 RD=OFF(1) WR=OFF(1) ALE=OFF(0) EA=OFF(1)

S101 RCS0=OFF FCS1=OFF RCS1=OFF FCS0=ON

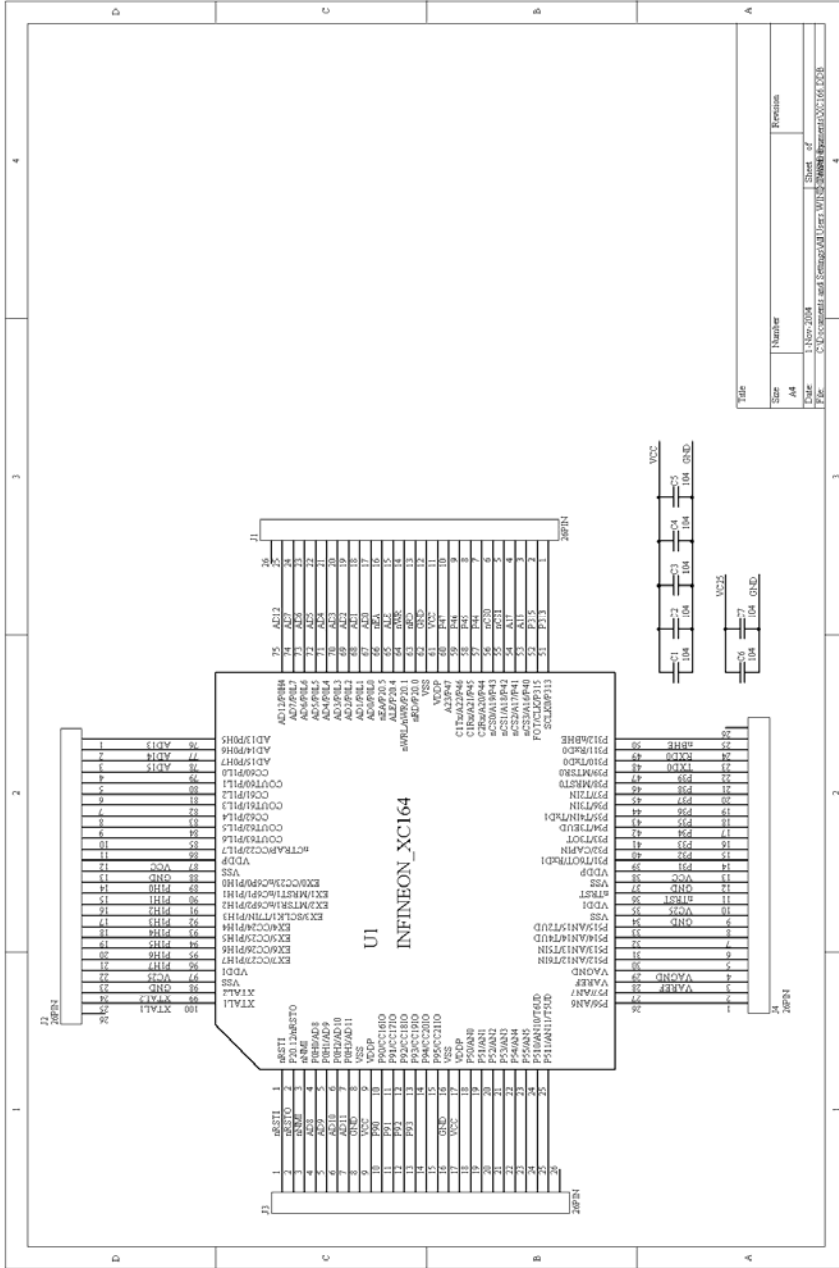
程序将 EX1AR 的程序写入到外部 FLASH AM29F800BA 中

下掉仿真器，将 S106、S101 配置为：

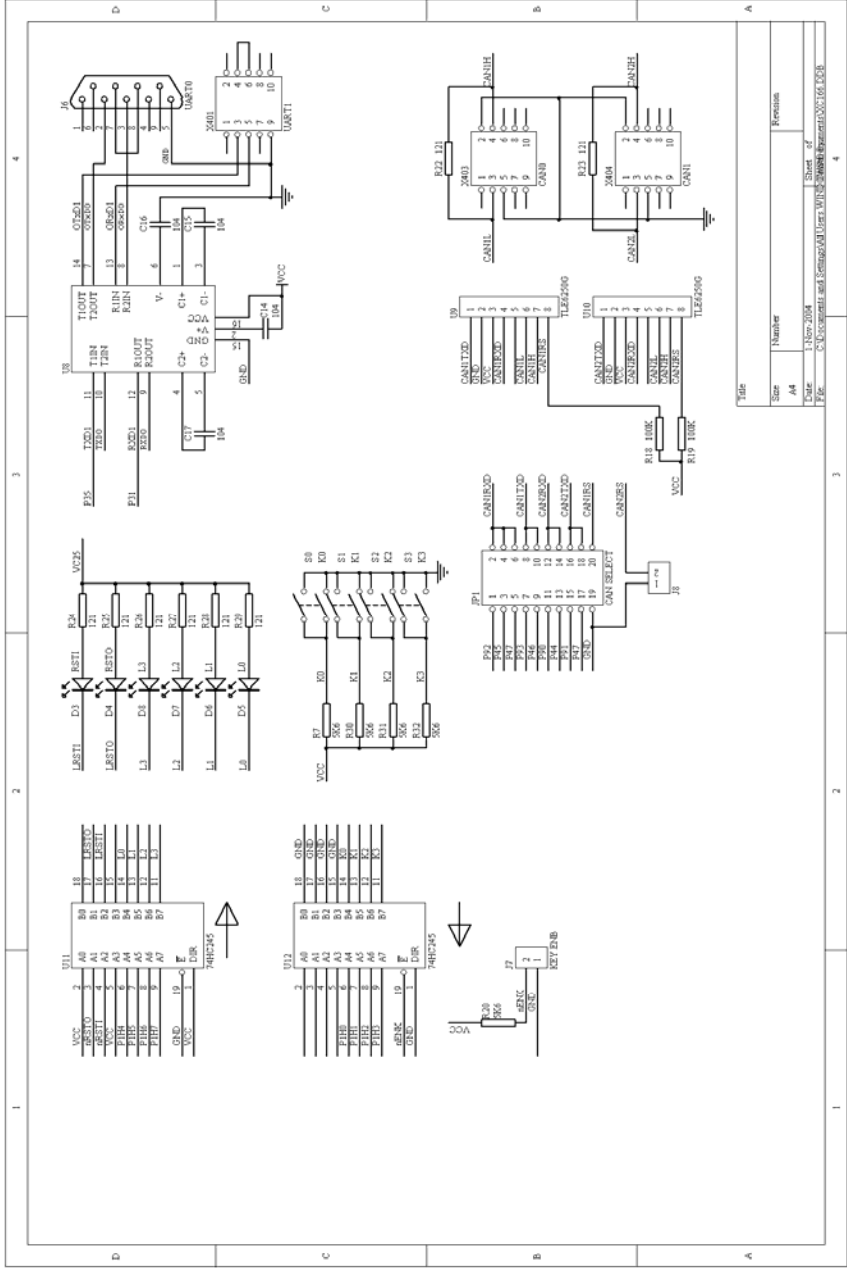
S106 RD=OFF(1) WR=OFF(1) ALE=OFF(0) EA=OFF(0)

S101 RCS0=OFF FCS1=OFF RCS1=OFF FCS0=ON

上电后，XC166 运行外部 FLASH 中的程序(EX1AR)，发光管闪烁。



Title		Size		Number		Revision	
A4		A4		A4		A4	
Date		1 Nov 2004		Sheet of		4	
File		C:\Documents and Settings\All Users\My Documents\Documents\XC164.XLS					



Title			
Size	Number	Revision	
A4			
Date	1-Nov-2004	Sheet of	
File		C:\Documents and Settings\All Users\My Documents\Projects\CAN\CAN.PCB	

1 简介

本手册介绍 Infineon XC866 用户板

XC866 指令兼容 INTEL MCS51, 时钟为两分频。

2 XC866 用户板功能

2.1 功能简介

- Infineon XC866 微控制器 (38 脚封装)
- 晶振插座
- 板上有一个 CAN 座、板上有一个 LIN 座、一个 ASC (RS232) 串口驱动、
- 一个 JTAG 口 (用于仿真)
- 八个 LED 发光二极管、三个按键输入

连接器

SUB-D9 串口连接器

16 脚双排针 (2x8) JTAG 口连接器

一个 10 脚双排针 (2x5) CAN 总线连接器

一个 2 脚单排针 LIN 总线连接器

两个 38 脚双排针 (2x19) XC866 管脚连接器

其它器件

一个电源指示灯 (5V)

复位按键

JP7 按键跳线器

JP13 LED 跳线器

CAN 允许跳线器

LIN 允许跳线器

UART (JP9) 管脚选择跳线器

2.2 JP5 跳线选择器

JP5 可以选择 MBC, TMS, P01, P21, P00, P20 管脚电平.

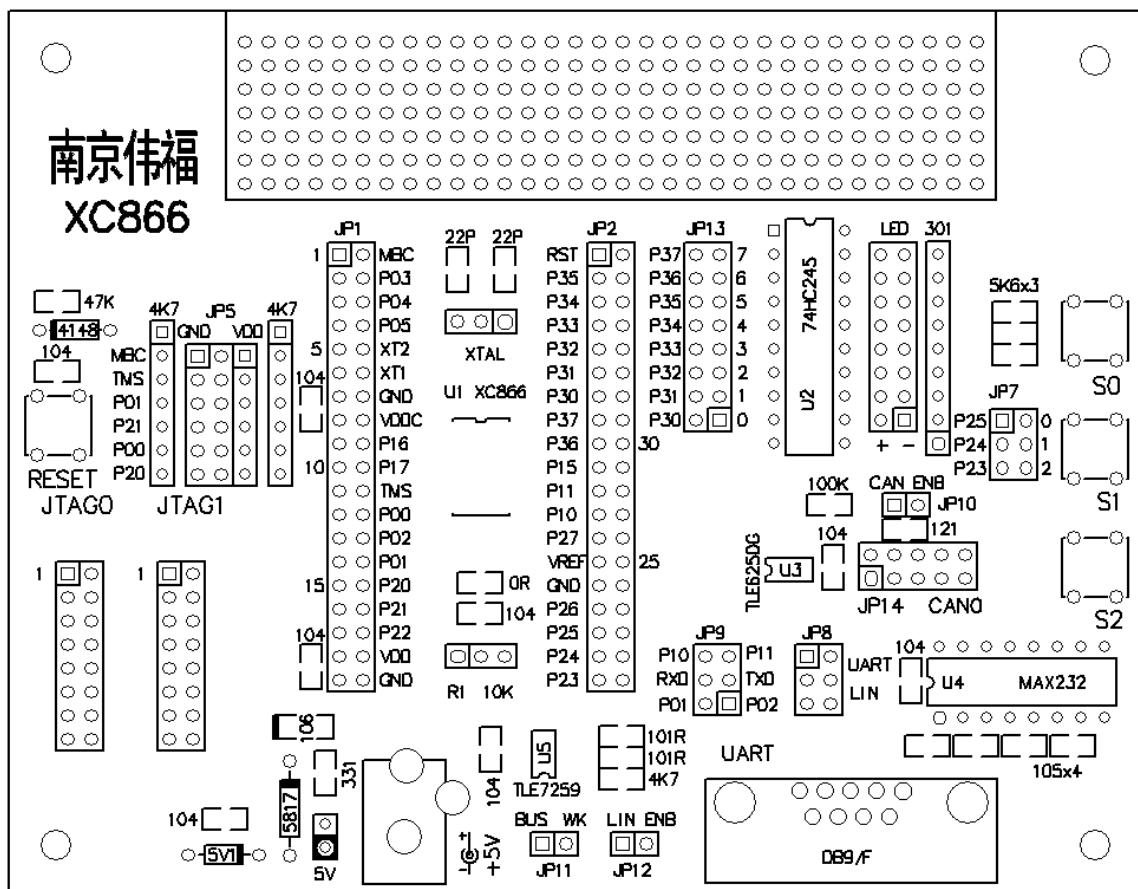
跳接 GND 时相应管脚对 GND 接 4.7K 下拉电阻.

跳接 VCC 时相应管脚对 VCC 接 4.7K 上拉电阻.

表一 JP5 跳线选择器

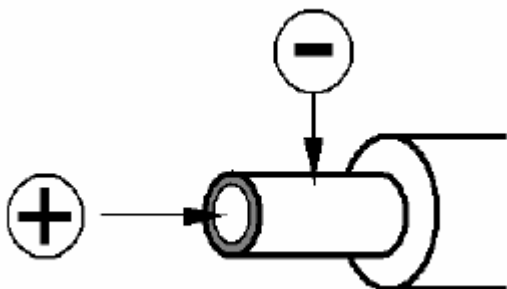
启动方式	MBC	TMS	P01	P21	P00	P20
用户方式	VCC	X	X	X	X	X
下载方式	GND	GND	X	X	X	X
OCDS 仿真	GND	VCC	X	X	VCC	X

2.3 用户板布局



2.4 供电

XC866 用户板使用单 5V 直流电源，电源插座内正外负，如图：



3. 缺省配置

XC866 用户板的缺省配置：

- JP13 全部短接，LED 发光管接 P3 口
- JP7 全部短接，S0, S1, S2 发光管接 P25, P24, P23
- JP10 不接 CAN 不工作，TLE6250 未焊接
- JP12 不接 LIN 不工作，TLE7259 未焊接
- JP9 接 P10, P11
- JP8 接 UART

4.1 连接器

4.1.1 RS232 连接器

4.1.2 CAN 连接器

4.1.3 LIN 连接器

4.1.4 JTAG0 连接器(接仿真器用)

4.1.5 JTAG1 连接器

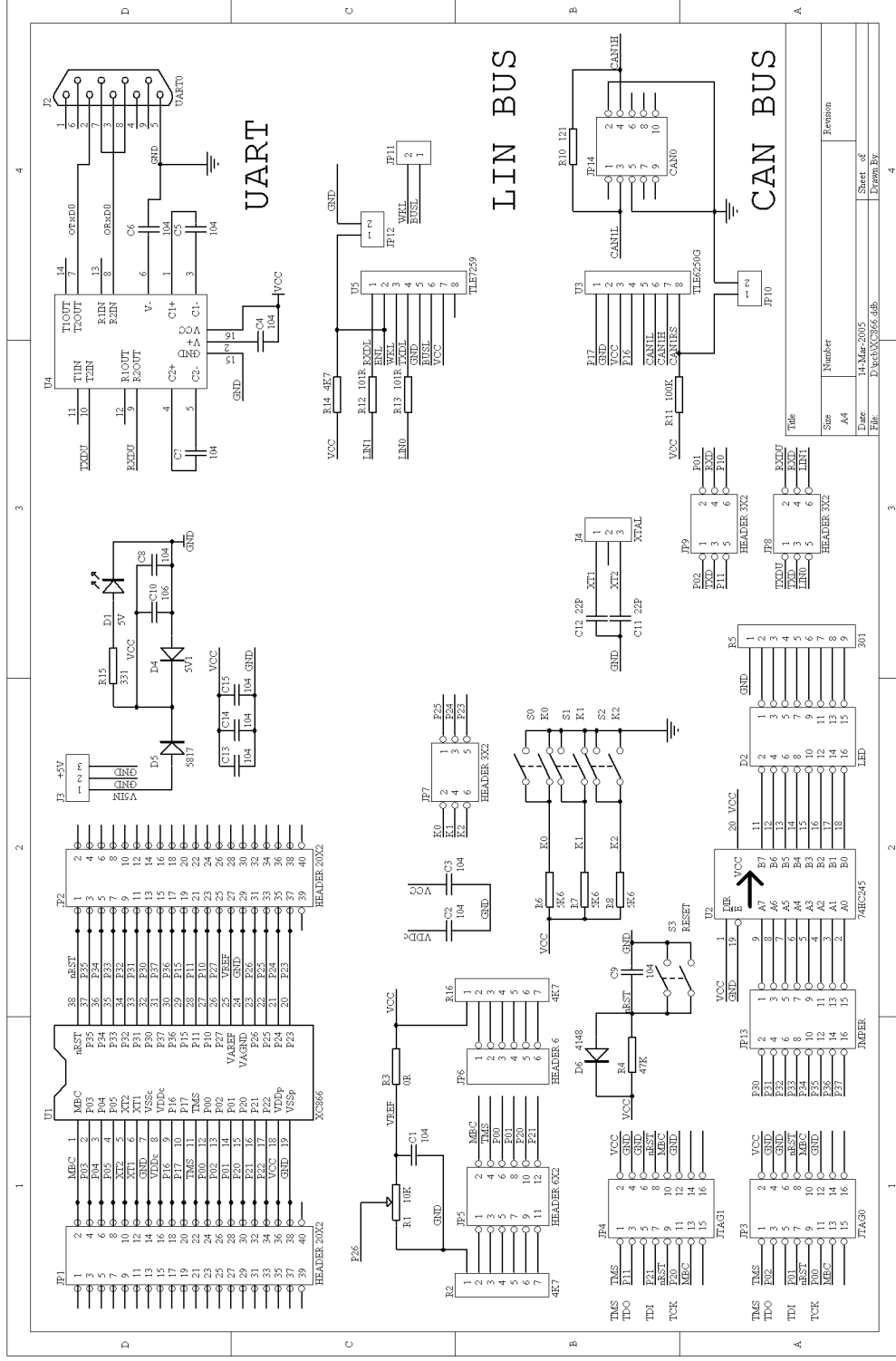
5 例子程序

在安装目录的 SAMPLES 目录下有 2 个例子程序，它们是：

EX1A.PRJ 发光管显示程序，用汇编语言编写。

EX1C.PRJ 发光管显示程序，用 C 语言编写。

6 XC866 用户板原理图



XC866 串口下载

XC866 的内部 FLASH 可以通过串口下载。WAVE866 软件含此功能。操作如下：

1. 在软件模拟方式下调入源程序并编译。或直接调入机器码 (BIN, HEX)
2. 按编程按钮。



3. 出现下面的编程窗口



选择串口号及波特率，按连接串口按钮。

4. 将 XC866 用户板上的 MBC, TMS 跳线器跳到 GND 端，并按一次复位按钮。此时，XC866 进如下载方式。
5. 按编程按钮进行编程

按钮功能

连接串口	:	连接 XC866 及 PC 机
擦除所有扇区	:	擦除 XC866 内部 FLASH 所有扇区
擦除使用扇区	:	擦除用户程序所占用的扇区
编程	:	将用户程序下载到 XC866 中

从 FLASH 0000 开始执行: 从 FLASH 0000 开始执行用户程序

从 XRAM F000 开始执行: 从 XRAM F000 开始执行用户程序